

AP20 Rec'd PCT/PTO 27 JUL 2006

System and method for communication between remote  
objects and local proxies

5

Description

The invention relates to a system and a method for communication between remote objects whose methods can be accessed as XML web services - in this case also referred to for short as web services - and their local proxy objects, whose methods implement transparent access to the remote methods. Communication methods such as these are described, for example, in <http://www.w3c.org/2002/ws/>, (Date: January 27, 2004).

15

XML web services represent a standardized and widely used basis for communication in loosely coupled, distributed systems, such as the Internet, in particular for calling remote procedures and methods.

20 XML web services can be used to access remote objects in object-oriented systems.

Figure 3 shows the system architecture of a communications system according to the prior art. This shows a client 1 which communicates with a service provider 10 via the Internet or a LAN 9. The client 1 can be regarded as a data processing device with associated software, with at least one client application 2 which communicates with proxies 3, for example a proxy A and a proxy B, with them in turn communicating with a communication layer 4. The communication layer 4 of the client 1 is designed to communicate with a corresponding communication layer 4 of a service provider 10 via the Internet 9. The facilities of the service provider 10 also include software means for provision of services 5, for example a service A and a service B.

The known procedure for a remote service call in an environment such as this is shown in Figure 3, and will be described in the following text:

- I. The client 1 calls a local procedure which is provided by a proxy 3.
- II. The proxy 3 uses the functionality implemented in the communication layer 4 to transmit the procedure call information via the LAN or the Internet 9 to the service provider 10.
- III. The procedure call information is extracted in the communication layer 4 of the service provider 10 and locally call the procedure 5 which implements that service, formats the return value of the call, and transmits this to the communication layer 4 of the client 1.
- IV. The communication layer 4 of the client 1 extracts the procedure return information and the corresponding local proxy 3 produces the return value as the result of the initial local procedure call to the client application 2.

By the use of SOAP (Simple Object Access Protocol), as described in <http://www.w3.org/TR/soap12-part1/>, linked to http (Hypertext Transfer Protocol), as described in <ftp://ftp.isi.edu/in-notes/rfc2616.txt>, as a method for interchanging structured and type-based data, services can be made accessible to widely different applications within the so-called WWW beyond platform boundaries. In principle, the known method as well as the method described below are, however, independent of the protocol being used.

One major disadvantage of a generic implementation of remote method calls such as this is that these calls are many times more time-consuming and computation-

intense than local function calls, or function calls in more closely coupled systems.

5 The invention is based on the object of specifying a system and a method for communication between remote objects, by means of which it is possible to reduce the communication complexity.

10 This object is achieved by a communications system which has the features specified in claim 1. An associated communication method and advantageous refinements are specified in further claims.

15 The invention accordingly proposes that the required communication complexity be reduced by arranging an optimization layer and a general proxy at the client end, and by introducing a general service at the service provider. These measures make it possible to achieve optimizations, for example by the grouping of  
20 calls.

The system or method allows the use of caches, alternative data formats and transmission protocols, as well as application-specific optimizations.  
25

In environments which, for example, only loosely couple individual components beyond the boundaries of a local area network, it may be particularly desirable to group service calls when the fixed costs for a so-called  
30 round robin from the client to the service provider and back are very high in comparison to the costs of the transmission and processing of the actual service request and response. These costs are caused by the necessity to set up (often several) connections, the  
35 creation and transmission of protocol-specific message headers, and the initialization of components which process messages (for example of SOAP/XML-Parser, authentication and authorization) or transmit messages

(for example HTTP connection or data compression). The invention results in an improvement in this situation since it allows optimizations to be introduced into the communication procedure independently of the creation  
5 of the services (web services) and proxies. In this case, there are no additional requirements on the procedure environment either of the service or of the client and proxy components, and the generic service and proxy components, in particular their interfaces,  
10 are not modified, thus advantageously achieving reusability and easy configurability.

The method according to the invention allows:

- 15 1. remote method calls to be suppressed, for example by using local caches for the response to information requests,
- 20 2. remote method calls to be delayed, for example the method calls which are required in object-oriented environments can be transmitted in order to monitor the lifetime of remote objects, whether this is by their explicit destruction (so-called destructor calls) or by  
25 the periodic determination of objects which are no longer being used (the so-called garbage collection), normally with a time delay from the client to the procedure environment of the remote object,
- 30 3. remote method calls to be grouped, for example in order to reduce the number of resource-intensive connection setting-up processes and to retrospectively deal with the subsequent requests in 1. and 2. when the opportunity arises,
- 35 4. the messages which contain the return values of remote method calls to have further information that is relevant for local use added to them, for example in order to cause the deletion of

entries which have become invalid in local caches,

- 5       5. used transmission protocols or data formats to be used to decide, depending on the propagation time environment, for example in order to replace the resource-intensive SOAP/HTTP transmission by a more efficient form, if the nature of the connection allows this (Internet/LAN, and if present HTTP proxies and  
10       firewalls), and
- 15       6. communication to be initiated independently of calls from client applications for management, in particular updating and invalidation of the data in a cache, or piggyback information together with the transmission of call groups and the reverse transmission of responses from the service provider to be requested.

The method can be used in all environments in which  
20       generic service provider components and proxy components are available which encapsulate the formalities for formatting and transmission of a remote service call, and the corresponding response. For example, components such as these are offered by the  
25       Framework Class Library (FCL) of Microsoft.NET Platform, see J. Richter, Applied Microsoft.NET Framework Programming, Microsoft Press 2001, pages 21 to 24.

30       A further description of the invention and of its advantages is provided in the following text with reference to one exemplary embodiment, which is illustrated in the drawing figures, in which:

35       Figure 1 shows the system architecture of a communications system according to the invention,

Figure 2 shows a communications procedure according to the invention, and

Figure 3 shows the system architecture of a communications system according to the prior art.

Figure 1 shows the system architecture of a communications system according to the invention. For communication monitoring and optimization purposes, a further web service with a more general functionality, which is referred to as general service 8, is installed at the data provided 10 and in addition to the services 5 which are provided in an embodiment environment according to the prior art, as already described above with reference to Figure 3. This general service 8 is able to switch one or more service requests to the original service provider, the services 5, and transmit one or more response messages to the remote client or clients 1.

A generic proxy for this additional web service is installed in a corresponding manner in the remote procedure environment, at the client 1, and is referred to as a general proxy 7. In addition, the client 1 contains an optimization layer 6, as an additional component. This offers communication and data formatting functions which are required for communication with the proxies 3, and to this extent replaces functions of the communication layer 4, but also with the addition of optimization capabilities. The optimization layer 6 may itself use the functionality of the communication layer 4, or some other implementation of the required functionality, by means of the general proxy 7.

Further information relating to the operation of the already mentioned components can be found in the following description of a communication procedure on

the basis of Figure 2. In Figure 2, vertical bars symbolize the respectively used system components, whose numbering matches the respective component designation in Figure 3. Procedure steps are annotated  
5 S1 to S14, and will be explained in the following text.

Figure 2 shows a typical procedure for a service call in an extended environment according to the invention:

Step S1: The client 1 or the client application 2,  
10 calls a local procedure which is made available by a proxy 3.

Step S2: The proxy 3 uses the functionality implemented in the optimization layer 6 for transmission of procedure calls. The optimization layer 6 offers the  
15 proxy 3 an interface which is identical to the original communication layer, for this purpose.

Step S3: Provided that it has not been delayed and cannot be answered by a result in a cache in the optimization layer 6, the service call has service  
20 calls which are already available but have been delayed or else which have been sensible to handle in advance added to it in the optimization layer 6 (grouping of calls).

Step S4: This group of calls is transferred by a local  
25 call of the general proxy 7 to the generic communication layer 4 of the client 1, and is transmitted to the communication layer 4 of the service provider 10.

Step S5: The group is transferred to the general  
30 service 8 by a local call to the general service procedure.

Step S6: The general service procedure processes the service calls and produces their results. The general service 8 finds the required services 5 for this  
35 purpose in the step 6.

Step S7: Method call by the general service 8.

Step S8: Result return from the services 5 to the general service 8.

Step S9: Combination of the results by the general service 8 to form groups.

Step S10: Return of the result group to the communication layer 4. The group can have information  
5 to be transmitted additionally from the service provider to the optimization layer 6 of the client added to it.

Step S11: Transmission to the remote communication layer 4 and return from there to the general proxy 7,  
10 which produces the result as a local procedure return to the optimization layer 6.

Step S12: Evaluation of the results in the optimization layer 6.

Step S13: The information which has been supplied to  
15 the initial service request in addition to the response is evaluated and used in the optimization layer 6.

Step S14: The response to the initial request is transmitted via the generic proxy 3 to the client application 2, thus completing this example of a  
20 procedure.

In one specific implementation of the system, for example as a component within the .NET framework, the extension to an existing system by the addition of  
25 generic proxies and service providers can be provided by replacement of the client-end communication layer by the optimization layer 6 according to the invention with a communication layer 4 and the installation of the additional general proxy in parallel with the  
30 existing proxies. Optimizations which are independent of the application, such as the use of caches and the delaying of service calls, can thus be made available in a configurable form, without any implementation effort. Application-dependent optimizations can be  
35 introduced as modules at a defined interface of the optimization layer 6.



The general service 8 is implemented in a corresponding manner in parallel with the available services at the service provider end. The general service 8 includes a cache of service-providing service instances, whose  
5 service methods are carried out by so-called reflection, that is to say on the basis of information about the destination, the name and the signature of the method. XML document instances in accordance with the SOAP specification or more bandwidth-efficient  
10 binary coding can be used for serialization and deserialization of the parameter and return objects relating to the method calls. In the case of SOAP (de)serialization, client-end types are associated with service-provider-end types on the basis of type  
15 attributes declared in the given proxy service pairs, and in the case of binary coding, a link is implemented between serialization and the type to be instantiated, on the basis of the requirements of the already found deserialization component.

20 The optimization and monitoring of the communication is transparent both for the user of the offered services and for their provider.